**Shell**

|  |  |  |
| --- | --- | --- |
| **SYNTAXE** | | |
| #!/bin/bash | Première ligne  Ou l’exécutable Perl, Python, Sh… | |
| #Commentaire | | |
| <cmd>… \  …<cmd2> | \ permet de fractionner une ligne en plusieurs | |
| Retour de 0 en cas de succès | | |
| "$VAR" au lieu de $VAR est une bonne pratique dans [ ]  ${VAR} dans des guillemets | | |
| . ./include.sh | Agit comme un include : On run ce script dans ce Shell | |
| **VARIABLES** | | |
| VAR=<Value> | VAR = <Value> ne marche pas : interprété comme une commande  Accès à la variable avec $VAR  Interprété comme une chaine de caractère  Utilisation d’une variable non assignée = variable vide "" | |
| **STRUCTURES CONDITIONNELLES** | | |
| If <cmd>  then  …  elif <cmd2>  then  …  else  …  fi | | Utilisation de la commande [ : Cf Bash\_CLI  Pour le résultat d’un test utilisable par if  If <cmd> ; then si on veut les mettre sur une même ligne  [ … ] && echo blabla , autre manière d’écrire un if else (&& ou ||) |
| Case $VAR in  Hello) cmd ;;  AutreTest) cmd ;;  \*) cmd ;;  esac | | \*) = défaut |
| **BOUCLES** | | |
| for i in <var1> <var2> … <varN>  do  Cmd  done | | Accès à i via $i |
| while <cmd>  do  Cmd  done | | while [ "$VAR" != "X" ] |
| while read f  do  …  done < myFile  Lecture d’un fichier, accès à chaque ligne via $f |
| While :  (= while true) |
| until <cmd>  do  …  Done | |  |
| select <var> in <var1> <var2> …  do  …  Done | | Affichage d’un menu en boucle.  Utilise $PS3 et $REPLY contient l’indice de l’élément pris |
| continue | |  |
| continue n | |  |
| break | |  |
| break n | |  |
| **TABLEAUX** | | |
| Tableau=(<var1> … <varN>) | | Création | écrasement  Echo ${tableau[i]} |
| Tableau[<indice>]=valeur | |
| **FONCTIONS** | | |
| Fonction(){  } | | Appel via fonction et non fonction() |
| function fonction{  } | |

**Généralités :**

* Appel de fonction : les $1… sont remplacés par leur nouvelle valeur, mais les variables globales d’avant peuvent être changées dans cette fonction.